Keywords: database management system integrity
biomedical data stream and signal processing

Michał WIDERA*, Janusz WRÓBEL*, Adam WIDERA*, Adam MATONIA*

A METHOD OF ENSURING DATA INTEGRITY IN A DATA STREAM MANAGEMENT SYSTEM

Assurance of data integrity is one of the prerequisites for each computer system. The paper presents a method enabling on-line maintenance of stream data set integrity. This method is implemented in a data stream management system prototype designed to find application in a biomedical monitoring system. In case of medical computer systems assurance of data integrity is particularly important for documenting formal results and for the patient's safety.

1. INTRODUCTION

In the case of a monitoring systems class, where the amount of information is large and the granulation is high, the application of a relational database management system is problematic. There are two basic obstacles that make it impossible to use the relational data management system; firstly, because of inadequate adaptation of the existing query language to the signal processing task and secondly, due to the implementation way of transaction mechanisms to ensure the consistency and integrity of a relational data set.

Assurance of data integrity is one of the prerequisites for each computer system [1,2]. If all kinds of system breakdowns are taken into consideration and all necessary repairs are done, the accumulated data set may be reconstructed. In case of medical computer systems this issue is particularly important for the justification of necessary formal results.

The loss of integrity of accumulated data poses a serious threat to every computer system. A breakdown may damage part of or a whole data set. In the case of medical computer system restart, distorted data can become a basis for incorrect interpretation and jeopardize the patient's health and life.

In case of a monitoring systems class, the differences can appear between archival and on-line recorded data as a result of quick and large data inflow. When analysing these differences it is necessary to foresee the divergence control in the project. The anomalies are caused by inappropriate design of recording and presentation procedures.

* Institute of Medical Technology and Equipment, 118 Roosevelt St, 41-800 Zabrze, POLAND
tel./fax: (32) 2716013/2712312, e-mail: michal@itam.zabrze.pl

2. RESOURCE REQUIREMENTS FOR MEDICAL SIGNAL PROCESSING

Data registration is a continuous process in the monitoring system. In this paper, we mainly concentrate on the registration of discreet signal problems. By a discreet signal we mean every data stream. The data stream can by described as a bag of elements, where the sequence of tuples is the first element and the second one is a function based on a set of natural numbers describing the time elapsed between consequent tuples in the stream [5,9].

A modern 12-lead ECG bedside monitor enables sampling of a biomedical signal at a rate of 1000 samples per second from each lead. Every sample represents a 16-bit number. The efficiency of the database management system is measured in transactions per minute. It can be proved that when omitting a protocol overhead, such an ECG medical monitoring system records data at a speed of about 700 000 samples per minute. The stream of data coming from just one medical device achieves intensity close to currently available modern and highly efficient relational systems (Table 1).

System	Database	tpm
HP Integrity rx5670 Cluster 64P	Oracle Database 10g Enterprise Edition	1,184,893
IBM eServer pSeries 690 Model 7040-681	IBM DB2 UDB 8.1	1,025,486
HP Integrity Superdome	MS SQL Server 2000 EE	1,008,144
HP Integrity Superdome	MS SQL Server 2000 EE 64-bit	786,646
IBM eServer pSeries 690 Turbo 7040-681	Oracle Database 10g Enterprise Edition	768,839
IBM eServer pSeries 690 Turbo 7040-681	IBM DB2 UDB 8.1	763,898

Table 1. Benchmarking of some leading database management systems. [11]

It needs to be remarked that in the case of an ECG signal we can alter the intensity of the recorded data stream. By reducing the granulation grade of the recorded information, i.e. by grouping data in samples (12 data items per sample) we can achieve a stream intensity of up to 60 000 tuples per minute. However, this is possible in case of synchronic signals, when we are sure that the data comes with an identical frequency in each stream.

Given the variety of recorded biomedical data and the need to compare on-line data from many data sources as well as benchmarking presented in table [1], we can state that the use of a relational database management system in a monitoring task is ineffective in case of on-line registration. Data analysis is ineffective, too.

3. THE METHOD OF ENSURING DATA SET INTEGRITY

Every transaction realizes a logically consistent set of operations. The system has to assure that the realization of every transaction will be completed with a success or failure. In case of a failure, the system has to assure that no changes will be made in the database. Managing this process in a multi-access environment constitutes one of the basic factors influencing the efficiency of the whole computer system. In 1983, T. Härder and A. Reuter [2] proposed the qualification of transaction properties set through the shortcut ACID [2], ambiguous but easy to memorize, meaning:

- **Atomicity** – each transaction constitutes an indivisible unit of processing. This property refers to the ability of the DBMS to guarantee that either all of the tasks of a transaction are performed or none of them are.
- **Consistency** – a database is in a consistent state if it is possible to obtain unambiguous and uniform answers from the information accumulated in a data set. The way of realizing the transaction should ensure the cohesion of the database. This property refers to the database being in a legal state when the transaction begins and when it ends.
- **Isolation** – a transaction happens independently of different operations including different transactions. The changes introduced in a data set are visible for other transactions only after the end of the transaction. This property refers to the ability of the application to make operations in a transaction appearing isolated from all other operations.
- **Durability** – in case of a system breakdown, the data set will not lose its consistency. This property refers to the guarantee that once the user has been notified of success, the transaction will persist and will not be undone.

Application of all ACID postulates for database users is connected with the impossibility of simultaneous realization of selected operations. In order to improve efficiency, the introduced requirements may be mitigated. Among others, the so-called levels of isolation [2] are introduced. As a result, users agreeing on some form of answer divergence obtain more effective solutions.

Research on distributed data management systems has been started in recent years. The arising problems required considerable mitigation of ACID postulates. The research on distributed databases conducted in the 1990s gave rise to Epsilon Serializability notion (ESR). In case of ESR, we can determine the acceptable degree of inconsistency for each Epsilon Transaction (ET). Various algorithms of divergence control (DC) for a centralized database system have been presented in literature [3]. However, ET has found the main application in distributed systems, where many processes work simultaneously in topologically distant places [4]. The whole work on ET forms the basis of the optimistic replication algorithms set. An additional advantage of ET is that it has a minor influence on the system's load. The issues connected with defining a stream transaction have not been widely reflected in literature so far [5,6,7,8,9,10].

4. CONTINUOUS MAINTENANCE OF STREAM DATA SET INTEGRITY

In search of an alternative data management method adapted to signal processing [12] we began working on a database management system based on data stream processing methods [12,13,14,15]. The implementation of all ACID requirements to a data stream set is problematic, even though we might think that it is necessary to create the transaction log and commit points management just like in the relational database systems. At the same time, while working with stream data we expect a high level of data granulation; therefore, the size of transaction log would increase even more rapidly than the base data stream. Additionally, after every minimal change of data set the commit point would cause a significant delay in every operation. Both results are undesirable; therefore, we have decided to develop our own method of continuous integrity maintenance.

The integrity of a data stream set is assured by the fulfillment of the following properties:

- **Consistency** – the end of the last transaction stands for the last integrity system state;
- **Atomicity** – each data stream appended to an operation is atomic;
- **Visibility** – each change of a data stream state is visible all processes at once;
- **Epsilon-Durability** – in case of a system breakdown, the last integrity data set state is restored.

By consistency we understand a stream set state that permits the reconstruction of a system state before a transaction. Due to a continuous unstable data stream set, this issue is realized by approximated states of a stream set computed "in past time". The matters of visibility and atomicity have no direct application to our model. Each operation of appending data is isolated and atomic. The stream state change is visible at once in all processes. The matter of Epsilon-Durability should have a similar meaning to that in a relational model and ET, i.e. the system breakdown should not influence the stream set consistency.

The potential damage during the appending process should not have any influence on the data set. However, system use buffer before operations are physically written. Buffers are flushed after overflow. This construction matters when considering the durability property. An example of this weakness appears when the system records streams with a high difference in stream intensities. Due to a lack of a stable state sequence, transactions should be released by an independent and parallel process. This process will flush buffers periodically.

When creating the principles of maintaining continuous integrity of streams data set in which the process of data accumulation should not be disrupted, it is necessary to present such a way of indicating scalar values that stream data appending processes would not be stopped.

5. A FORMAL METHOD OF ENSURING STREAM DATA SET INTEGRITY

By a biomedical monitoring system we understand a computer system connected to monitoring devices. The streams sources are independent. We assume that each monitoring device generates a data stream [12,13]. A computer system collects, stores and analyzes information in a database management system and then presents it to medical personnel in a form legible for humans. The system can replay and reanalyze the recorded information on direct request.

Figure 1 presents the data flow in such a system. Data are collected in buffers processed inside a biomedical monitoring system. The process of data gathering in a monitoring system is asynchronous.

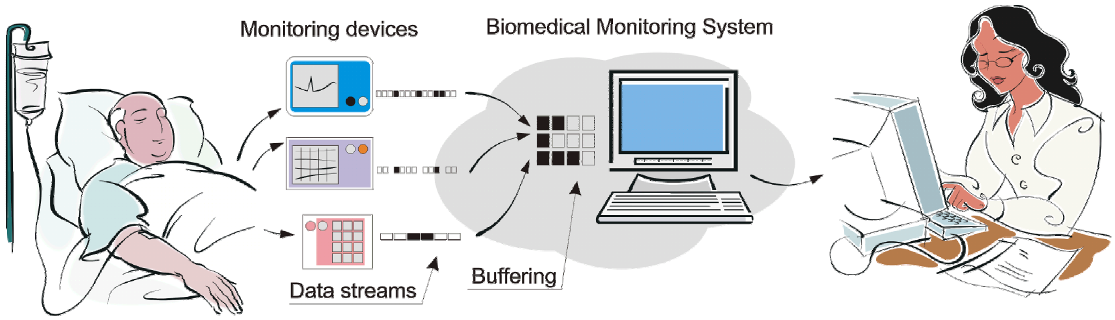

Fig.1 Data flow in monitoring system

By a system state we mean the set of all information collected by the system in time periods. By a stable stream set we understand the appropriate information about the length and data content of the stream. An approximated state of a stream stands for computed information about the expected stream length and data content in a given period of time.

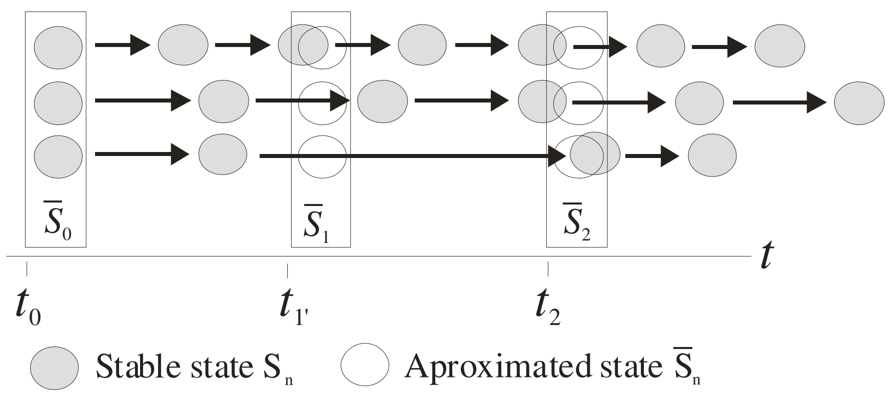

Fig.2 Transactions over data streams

Figure 2 presents parallel processes of continuous data append-only of three separate streams. Stable states are marked in grey. The process was presented using a timeline. The time of synchronization of buffers t_n is periodic and the period is predefined. Management of stream and signal transaction processing is one of our current research goals.

The sequence of approximated states of a stream set is described by index. By index we understand a sequence of vectors whose elements are characterized by a computed length of each data stream in time. During the construction process of an index the system may temporarily lock the recording of data. However, our main goal is to prevent it from happening. A stable state of a stream is represented by vector (1):

$$\overline{S_0} = [\{s_1\}_n, \{s_2\}_n, \ldots, \{s_m\}_n] \quad (1)$$

Each element of vector (1) is a sequence of tuples. When describing successive vector $\overline{S_n}$ we need to describe transition vector $\overline{\tau_n}$. Transition is a set of operations realized over a data set that enables changing one state of a data set into another. $\overline{\tau_n}$ is related with $\overline{I_n}$. This index can be presented as vector $[i_1, i_2, \ldots, i_m]$ where each element i_k ($k = 1..m$) contains a length of sequence $|\{s_k\}_n|$ measured in the same time t. Considering parallel and undisturbed work of append-only processes, we cannot simultaneously measure the length of $|\{s_k\}_n|$ and $|\{s_{k+1}\}_n|$. Therefore, we need to create a sequence of vectors containing approximated values. Each value $|\{s_m\}_n|$ is based on approximated measurements. Thus, each state $\overline{S_n}$ will be approximated, too. Figure 2 presents this process graphically.

The proposed solution is as follows: Not blocking the measure of value $|\{s_m\}_n|$ in time t executed for each stream m represents value $l_{m,t} = \left|\{s_n\}_m\right|_t$. After one measure of a data set, we obtain vector (2) of pairs:

$$\bar{q} = \left[\left(l_{0,t_0}, t_0\right), \left(l_{1,t_1}, t_1\right), \ldots, \left(l_{m,t_m}, t_m\right)\right] \quad (2)$$

When approximating by mean average $\overline{S_n}$ values in time t we need to determine another vector (3)

$$\overline{q'} = \left[\left(l'_{0,t_0}, t'_0\right), \left(l'_{1,t_1}, t'_1\right), \ldots, \left(l'_{m,t_m}, t'_m\right)\right] \text{ where } \underset{i}{\forall} t_i < t_i' \quad (3)$$

and each $l'_{m,t} = \left|\{s_n\}_m\right|_t$ is approximated, too.

It needs to be remarked that the above method is closer to mechanisms applied in journaling file systems than to transactions in database management systems.

6. DIVERGENCE CONTROL

Considering that the consequent states of data set are approximate we expect that system answers may differ depending on the moment of issuing queries. The question of divergence control over data stream management system [4,13,14] must be investigated. By divergence control, we understand a difference in system answers related to the same data, but issued ad-hoc and in relation to archival data.

Systems analyzing continuous data inflow use applied algorithms prepared to support answers for incomplete or disturbed data. Thus, the divergence of answer, depending on when the query was done, is a feature of such a system. A system which records the content of previously given answers has an easier task.

However, if the order of incoming data is recorded with a given probability, the system is unable to retrieve the actual order of events. The proposed transaction mechanism makes it possible to estimate the divergence when the input stream is retrieved. Retrieving the real order of events is not possible because the length of streams cannot be measured without stopping the append-only processes.

7. CONCLUSIONS

The integrity of recorded data must be assured due to formal and legal requirements. The easiest way to do it is to assure integrity in a database management system. Unfortunately, the use of a relational database management system for on-line registration and data analysis is ineffective. Hence, in search of an alternative solution, we began working on a new database management system based on the concept of a data stream management system.

The issue of ensuring data stream integrity has not been widely discussed in literature so far. Therefore, we need to find our own method of maintaining integrity. If all ACID postulates are secured for database users, it is impossible to realize many operations at a time. Therefore, to improve efficiency, this method mitigates the introduced ACID requirements. The presented method enables the maintenance of stream data set integrity on-line. This method is implemented in a prototype of a data stream management system designed to find application in the monitoring system of biomedical processes.

ACKNOWLEDGEMENT

This study is financed by the State Committee for Scientific Research resources in 2003-2005 years as a research project KBN Grant No. 4 T11F 002 25.

BIBLIOGRAPHY

1. HÄRDER T., REUTER A.: Principles of Transaction-Oriented Database Recovery, IBM Research Report RJ 4214, San Jose, Calif., in: Computing Surveys, Vol. 15, No. 4, Dec. 1983, pp. 287-317.
2. PANKOWSKI T., Więzy integralności a zarządzanie transakcjami w SQL-owych bazach danych, Informatyka 1995, Nr 9, s. 21-26.
3. WU K.L., YU P.S., PU C., (1992) Divergence control for epsilon-serializability in Proc. of Int. Conf. on Data Engineering : 506-515
4. PU C., HSEUSH W., KAISER G. E., WU K.L., YU P.S. (1995) Divergence Control for Distributed Database Systems, Distributed And Parallel Databases, 3 : 85-109
5. BABCOCK B., BABU S., DATAR M., MOTWANI R. WIDOM J. Models and Issues in Data Stream Systems, Invited paper in Proc.of the 2002 ACM Symp. on Principles of Database Systems (PODS 2002), June 2002
6. L. GOLAB, M.T. OZSU: Issues in data stream management, ACM SIGMOD Record Volume 32, Issue 2, pp.5-14, June 2003
7. ARASU A. BABU S. WIDOM J. An Abstract Semantics and Concrete Language for Continuous Queries over Streams and Relations, Technical Report, November 2002.
8. ABADI D., CARNEY D. CETINTEMEL U. CHERNIACK M. CONVEY C. LEE S. STONEBRAKER M. TATBUL N. ZDONIK S. Aurora: A New Model and Architecture for Data Stream Management. In VLDB Journal, August 2003
9. MOTWANI R. WIDOM J. ARASU A. BABCOCK B. BABU S. DATAR M. MANKU G. OLSTON C., ROSENSTEIN J. VARMA R.. Query Processing, Resource Management, and Approximation in a Data Stream Management System In Proc. of the 2003 Conference on Innovative Data Systems Research (CIDR), January 2003
10. ABADI D. , CARNEY D. , CETINTEMEL U. , CHERNIACK M. , CONVEY C. , LEE S. , STONEBRAKER M. , TATBUL N. , ZDONIK S. : Aurora: A New Model and Architecture for Data Stream Management. VLDB Journal, vol .12, nr. 2, pp. 120-139, 2003
11. POESS M., FLOYD C.: New TPC Benchmarks for Decision Support and Web Commerce, ACM SIGMOD Record, 29(4) December 2000 , http://www.tpc.org/.
12. WIDERA M., J. WRÓBEL J.,HOROBA K.,JEŻEWSKI J.,KUPKA T.,WIDERA A.: Filtracja zakłóceń w strumieniowym systemie przetwarzania sygnałów, Współczesne problemy sieci komputerowych – Nowe Technologie, WNT 2004, pp. 367-377
13. M. WIDERA, J. JEŻEWSKI, R. WINIARCZYK, J. WRÓBEL, K. HOROBA, A. GACEK: Data stream processing in fetal monitoring system: I. Algebra and query language, Journal of Medical Informatics & Technolo-

gies, vol .5, pp. 83-90, 2003

14. WIDERA M. WINIARCZYK R. JEŻEWSKI J. GACEK A. WRÓBEL J. K. HOROBA: Strumienie danych w systemie monitorowania, analizy i klasyfikacji sygnałów biomedycznych, STUDIA INFORMATICA Volume 24, Number 3 (55), pp. 35-45, Gliwice 2003,
15. JEŻEWSKI J. WRÓBEL J. HOROBA K. Monitorowanie zagrożeń płodu wspomagane komputerem „Biocybernetyka i Inżynieria Biomedyczna 2000 – Tom 7. Systemy komputerowe i teleinformatyczne w służbie zdrowia" red. M. Nałęcz, Akademicka Oficyna Wydawnicza EXIT, pp. 97-119, Warszawa, 2001

Reference: M.Widera, J.Wróbel, A.Widera, A.Matonia: Method of the ensuring data integrity in the data stream management system, Journal of Medical Informatics & Technologies Vol. 8 October 2004 ISSN 1642-6037, 2004, pages. 141-148